REMARKS

In the final office action, claims 1, 3-5, 9, 12, 17 and 18 are rejected under 35 U.S.C §102(e) as being anticipated by U.S. Patent No. 6,801,536, to Foster et al (hereinafter "Foster et al"). Claims 2, 10 and 19 are rejected under 35 U.S.C §103(a) as being obvious over Foster et al in view of U.S. Patent No. 5,732,324, to Rieger III (hereinafter "Rieger III"). Claims 6, 7, 13-15, 20 and 21 are rejected under 35 U.S.C §103(a) as being obvious over Foster et al in view of U.S. Patent No. 5,815,671, to Morrison (hereinafter "Morrison"). Claim 11 is rejected under 35 U.S.C §103(a) as being obvious over Foster et al in view of Rieger III and Morrison. Claims 8 and 16 are rejected under 35 U.S.C §103(a) as being obvious over Foster et al in view of Morrison and U.S. Patent Application Publication No. US 2003/0212996, to Wolzien (hereinafter "Wolzien").

The Examiner states that Foster et al teaches the present invention as claimed because the system therein "creates 'subblocks prior to transport' and therefore teaches data files partitioned into segments interspersed in a broadcast signal." As discussed in more detail below, the Examiner cannot arbitrarily switch between the packets in the transport stream and the subblocks built at the set-top box to state which aspects of the system disclosed in Foster et al purportedly teach the recited elements in the claims. If the packets in the transport stream of Foster et al could arguably be considered data files as claimed, then the recitations in claim 1 regarding storing and monitoring progress of segments of a data file cannot be met by the processing of the transport data stream packets at the STB. Further, the subblocks cannot be argued to teach segments as claimed since they are created at the STB and therefore after transport.

The office action apparently analogizes packets in a transport stream of Foster et al to be data files as claimed and bytes in those packets to be segments as claimed. This is incorrect since the bytes of a packet are not interspersed in the transport stream

of Foster et al, in contrast to segments of the data file recited in claim 1 being interspersed in a broadcast signal. Secondly, the buffers 222A and 222V in Foster et al do not monitor progress of storage of bytes in a particular packet, nor even the progress of storing packets. Mere sizing of a buffer to control when it is filled and when it is read from does not constitute monitoring progress of reception of bytes associated with a packet. Also, it is impermissible to change the apparent analogy in the office action to purport that a packet in Foster et al teaches a segment as claimed, rather than a transport packet byte, since there is no discussion in Foster et al of relating the packets to a particular data file that has been partitioned into packets. For example, unlike the claimed invention, there is no disclosure or suggestion in Foster et al of a file being segmented into a plurality of packets, and a header being provided in the transport stream to indicate the number of packets that constitute that file. The headers of the transport stream packets of Foster et al merely identify the type of data (e.g., audio or video) in the packet, but do not relate the packets to others that constitute a data file, unlike the claimed segments and header

Finally, the subblocks or data blocks described in Foster et al do not disclose or teach segments as claimed. There are numerous passages throughout the text of Foster et al that describe defining and building subblocks and data blocks at the set-top box (STB) and therefore *after transport and after reception* of a transport stream. The STB receives a transport stream 210 (e.g., packets of 188 bytes including a four byte header) at demultiplexer 110, and converts it to a packetized elementary stream format (PES) including separate audio and video streams that are provided to separate queues (see column 3, line 61 to column 4, line 2 and column 4, lines 38-41). A byte-to-interrupt (BTI) signal determines when subblocks are formed *in the* STB and sent to a queue in the STB (see column 6, lines 1-49). This is most plainly evident from Fig. 3 of Foster et al, where it is clearly shown that the transport stream is *received* (block 310), *and then* the subblocks are defined via queues for audio and video (block 320), *and then* the sub block headers are built (block 340), as described at column 5, line 43 through column 6, line 49 of Foster et al. See also Fig. 1 where a transport

stream is input to a transport demux 110 of a STB chip 100. See also Fig. 2 where the transport demux 220 of the STB chip has audio and data buffers 222A and 222V. As stated at column 6, lines 23-25 of Foster et al, a bytes-to-interrupt (BTI) signal is a "rolling interrupt that indicates when a given number of bytes (in 256 byte increments), *forming sub-blocks* (emphasis added) have been sent to a queue." Thus, the subblocks are created at the STB chip 100 and therefore after transport. See also column 5, lines 56 and 57 which states that the transport packets *are received* in the correct order which enables the buffers 222A and 222V to be in the transport demultiplexer of the STB (i.e., transport demultiplexer 220 in Fig. 2 or transport demultiplexer 110 in Fig. 1). See column 6, lines 38-42 of Foster et al which state "Upon the issuance (330 of Fig. 3) of each BTI interrupt and as a data block is sent to multiplex buffer 280, a *header is created* (emphasis added) for each sub block...." The "information for the header is *developed concurrently* (emphasis added) with the storage of data subblocks to the multiplex buffer 280. As stated at column 6, lines 45-49 of Foster et al, sub block and their headers are read out from buffers by queuing for storage. Thus, all of these receiving, buffering/queuing, sub block and header building operations are performed *at the STB after the transport stream* 110 (Fig. 1) or 210 (Fig. 2) *is received.* See also the claims of Foster et al. Claim 2 of Foster et al recites steps for defining subblocks using BTI values and queuing controlled by same. Claim 5 recites building a header after buffering and defining subblocks. To interpret the transport stream to the STB in Foster et al as already having subblocks prior to transport is contrary to the disclosure in Foster et al.

In the office action, the Examiner is using text and selected words in Foster et al out of context and therefore impermissibly to purportedly teach the claimed invention. For example, column 3, lines 65-67 of Foster et al relied on by the Examiner merely describe how the MPEG transport stream 110 is converted to a compressed and encoded packetized elementary stream (PES) format by the *STB* chip 100 (see column 3, lines 61 and 62). As stated in column 4, lines 38-43, the *STB* (emphasis added) separates audio and video data from the transport stream into two

packetized elementary streams (PESs) and the granularity of the PESs is coarser than the relatively small packet size of the transport stream.

The office action also references column 4, lines 55-67 and column 5, lines 43-67 of Foster et al to purportedly teach a transmission signal made up of packets having a header as claimed. First, as stated above, the packets and packet bytes of the transport stream of Foster et al bear no analogy to the recited data files and segments. Second, the text at column 4, lines 55-67 of Foster et al has been mischaracterized in the office action. The text merely states how a transport packet header can indicate the number and correspondence of transport stream bytes in transport stream packets to packets *in PES steams created at the STB* (e.g., the PES packets are large compared to the size of the transport packets). The PESs, however, are created at the STB as stated above and therefore cannot teach or suggest a data file as claimed.

The office action refers to column 5, lines 43-55 to purportedly teach the data files, segments and headers as recited in claim 1. This text, however, refers to interspersing transport stream packets into PES streams that are *created at the STB*, and therefore does not teach or suggest segments interspersed in a broadcast stream *prior to transport*.

In view of the above, Foster et al fails to teach or suggest the invention recited in claims 1 and 17. Further, regarding claim 3, the office action appears to suggest that packets in Foster et al teach data files as claimed and that bytes in the packets teach segments as claimed. This is incorrect for reasons stated above. In addition, there would be no need for a header as recited in claim 3 since the packet size is known (i.e., 188 bytes including a four byte header). Further, Foster et al does not teach providing a transport stream packet with data indicating the size of the buffers 222A and 222V that store the transport packets. In addition, the storage area for the subblocks and data blocks also does not teach in the invention recited in claim 3 since the sub blocks are built at the STB from PES using BTI signals and therefore have no relation to a packet in the transport stream which the Examiner purports to be a data

file as claimed. The size of these data blocks is determined at the STB (see column 6, line 1 to column 7, line 8). Thus, Foster et al does not teach the invention recited in claim 3. In addition to the reasons stated for claim 3, Foster et al does not teach or suggest the invention recited in claim 4 (e.g., unique codes for the packets or packet bytes in the transport stream, nor the first and second fields recited therein), and therefore does not teach or suggest claim 5 which depends from claim 4.

Regarding claims 9 and 18, Foster et al does not teach or suggest the STB determining which packets or bytes in packets have not been received. The text at column 5, lines 43-67 of Foster et al merely states that the transport packets are received in order, and is silent as to what happens if a transport packet is dropped.

The office action states that the queuing buffer system disclosed in Foster et al purportedly teaches the invention recited in claim 12, that is, first and second portions of memory for storing, respectively, complete data files and data files for which segments are still being received. This is incorrect. First, the buffers relating to the subblocks and data blocks (e.g., buffers 240A, 240B and 280 in Fig. 2 of Foster et al) cannot teach or suggest the portions of memory as claimed since they store blocks of data *defined and built at the STB* and not a data file in the transport stream. Secondly, Foster et al is silent regarding when a transport packet or byte therein is not received at the buffers 222A and 222V.

With regard to claim 18, the text at column 8, lines 15-35 in Foster et al and relied on in the office action refer to merely monitoring for when a buffer is full to initiate a transfer. The buffer in question is for data blocks *created at the STB* which are not the packets or packet bytes in the transport packet stream that have been apparently compared in the office action to the claimed segments of data file in a broadcast stream. Further, dumping a buffer when it is merely full does not teach monitoring for *which* segments in a selected data filed have not yet been received as claimed.

Rieger III is relied on for its purported disclosure of alerting a user when data segments have been stored in a memory device as claimed in claim 2. Applicants respectfully submit that Rieger III does not disclose such alerting, and that the Rieger III does not overcome the deficiencies of Foster et al. Rieger III teaches sending audio programs from low power transmitters to proximate digital burst radio (PDBR) receiving units in motor vehicles. The programs have preambles identifying programs by a brief textual description and date or creation. Thus, Rieger III does not teach a header comprising information indicating the number of said segments that constitute a data file. Rieger III merely teaches that a receiving unit can use the preamble to filter previously received programs based on the brief textual description and date of creation in the preamble. Rieger III, however, cannot use the preamble to "monitor the progress of storage of said segments" as recited in claim 1. Rieger III merely teaches determining if an entire program is received and stored, and not its progress.

Morrison is relied on for its purported disclosure of message data codes in sent data. Applicants respectfully submit that Morrison does not overcome the deficiencies of Foster et al. Morrison does not teach or suggest a data file characterized in a broadcast signal by a header indicating the number of segments that constitute the data file, among other aspects of the claimed invention. Further, the STC in Foster et al discussed in the Office Action with respect to claim 13 is added at the receiver and therefore does not suggest a segment header in a broadcast signal as claimed.

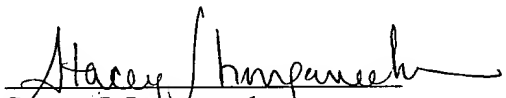
Paragraph [0058] of Wolzien is relied on for its purported disclosure of code identification information that identifies a type of car for a user profile to facilitate an automated push information operation. Applicants respectfully submit that Wolzien does not overcome the deficiencies of Foster et al. Further, none of these three references singly or in combination teaches or suggests the invention recited in claims 1 or 13, the base claims from which claims 8 and 16 depend for reasons set forth above. For example, Wolzien does not teach or suggest partitioning of a data file in a broadcast signal into segments and providing headers in the broadcast signal to

indicate the number of segments in a data file, as recited in both of claims 1 and 13. Applicants therefore respectfully request withdrawal of this basis for rejecting claims 8 and 16 under 35 U.S.C §103(a).

In view of the foregoing, Applicants respectfully request withdrawal of the 35 U.S.C §§ 102 and 103 rejections of the claims 1-21 set forth in the Office Action.

In view of the above, it is believed that the application is in condition for allowance and notice to this effect is respectfully requested. Should the Examiner have any questions, the Examiner is invited to contact the undersigned at the telephone number indicated below.

Respectfully submitted,



Stacey J. Longanecker
Attorney for Applicants
Reg. No. 33,952

Roylance, Abrams, Berdo & Goodman, L.L.P.
1300 19th Street, N.W., Suite 600
Washington, D.C. 20036
(202) 659-9076

Dated: 31 October, 2005